

Languages recognized by nondeterministic quantum finite automata^{*}

Abuzer Yakaryılmaz and A.C. Cem Say

Boğaziçi University, Department of Computer Engineering,
Bebek 34342 İstanbul, Turkey
abuzer,say@boun.edu.tr

February 12, 2010

Abstract. The nondeterministic quantum finite automaton (NQFA) is the only known case where a one-way quantum finite automaton (QFA) model has been shown to be strictly superior in terms of language recognition power to its probabilistic counterpart. We give a characterization of the class of languages recognized by NQFA's, demonstrating that it is equal to the class of exclusive stochastic languages. We also characterize the class of languages that are recognized necessarily by two-sided error by QFA's. It is shown that these classes remain the same when the QFA's used in their definitions are replaced by several different model variants that have appeared in the literature. We prove several closure properties of the related classes. The ramifications of these results about classical and quantum sublogarithmic space complexity classes are examined.

1 Introduction

An interesting feature of both probabilistic and quantum computational models is that in some cases, the set of problems that can be solved gets larger when the automaton in question is allowed to make more error in its decisions, whereas in some other cases, such a relaxation does not increase the computational power at all. When one-way probabilistic finite automata (PFA's) are required to make no error in their decisions, they recognize exactly the class of regular languages. When they are allowed to make *bounded error*, that is, to give the correct response for each input with probability at least $\frac{1}{2} + \delta$, for a fixed $\delta > 0$, the class of languages that are recognized remains the same. The computational power of PFA's is seen to increase only when we allow two-sided *unbounded error*, where the only requirement is that all members of the recognized language are accepted with probability greater than the acceptance probability of any nonmember.

There are several alternative models of quantum finite automata (QFA's), and differences (e.g. in features regarding the form and number of measurements that can be performed on the machine, whether an output tape is included or not, etc.) in their definitions, which would not affect the power of classical automata, yield QFA variants of differing computational power. In the zero-error and bounded-error cases, some QFA variants [28,20,30] are strictly inferior to the corresponding PFA's from the point of view of language recognition power, whereas the most general models [31,11,7,17] are equivalent to their probabilistic counterparts in those settings. In the unbounded error case, the languages recognized with cutpoint by the weakest QFA model [28] form a proper subclass [6] of the corresponding classical class (the stochastic languages), whereas it was discovered recently [42,43] that more generalized QFA variants, including the popular Kondacs-Watrous model, are equivalent to PFA's in this case as well.

^{*} This work was partially supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) with grant 108142 and the Boğaziçi University Research Fund with grant 08A102.

With regard to state complexity, sufficiently general QFA models can simulate all zero-error and bounded-error PFA's with small overhead, and some regular languages have bounded-error QFA's that are exponentially smaller than the corresponding PFA [3]. In the two-sided unbounded error setting, quantum and probabilistic machines can simulate each other with only a polynomial overhead in the number of states [43].

We study the computational power of QFA's in the one-sided unbounded error setting, where one of the two responses that the machine can output about the membership of the input string in the recognized language is correct with certainty, and the other response has a nonzero probability of being correct. Since the error bound can be improved by repeating the computation, an examination of languages recognizable in this setting is significant for understanding the power of generalizations of the underlying model to include, say, a two-way tape head. Just like their classical counterparts, QFA's that recognize their languages with cutpoint 0, (that is, with positive one-sided error,) are also known as *nondeterministic* machines. It is well known that classical nondeterministic finite automata recognize precisely the regular languages. In notable previous work on nondeterministic quantum finite automata (NQFA's), Bertoni and Carpentieri have shown [6] that the class of languages recognized by NQFA's of the Moore-Crutchfield type does not contain any nonempty finite languages, but does contain the nonregular language $L_{neq} = \{w \in \{a, b\}^* \mid |w|_a \neq |w|_b\}$, where $|w|_\sigma$ denotes the number of occurrences of the symbol σ in the string w . Nakanishi et al. [29] considered the somewhat more powerful Kondacs-Watrous model of QFA's, and proved that NQFA's of this type can recognize all regular languages, establishing their strict superiority over their classical counterparts.

In this paper, we give a full characterization of the class of languages recognized by all NQFA variants that are at least as general as the Kondacs-Watrous type, demonstrating that it is equal to the class of exclusive stochastic languages. This lies properly between the classes of languages recognized with zero error and two-sided unbounded error by QFA's¹. Every regular language has a NQFA with at most linearly more states than the corresponding classical nondeterministic finite automaton (NFA), and there exist infinite families of regular languages which can be recognized by just tuning the transition amplitudes of a NQFA with a constant number of states, whereas the sizes of the corresponding NFA's grow without bound. We also prove several new closure properties of the related classes, and examine what these results imply about the comparative power of probabilistic vs. quantum Turing machines with small space bounds.

The rest of this paper is structured as follows: Section 2 contains the relevant definitions and previously known facts. In Section 3, we give a characterization of the class of languages recognized by Kondacs-Watrous NQFA's, and discuss the superiority of several NQFA variants over their classical counterparts in terms of language recognition and succinctness. An examination of the relationships among languages which can be recognized by QFA's with one-sided error and those that require two-sided error is presented in Section 4. Section 5 contains several proofs of closure properties for the classes of languages recognized with one-sided error. Section 6 is a conclusion, where we examine the consequences of the NQFA results for classical and quantum sublogarithmic space complexity classes.

¹ To our knowledge, this is the only case where these three classes have been shown to be distinct for any automaton model, be it quantum or classical.

2 Preliminaries

2.1 Automata

In the following, Σ denotes the input alphabet, not containing the end-markers \mathfrak{c} and \mathfrak{s} , and Γ is the tape alphabet, such that $\Gamma = \Sigma \cup \{\mathfrak{c}, \mathfrak{s}\}$.

Definition 1. A (1-way) probabilistic finite automaton (PFA) with $n \in \mathbb{Z}^+$ states is a 4-tuple $\mathcal{P} = (S, \Sigma, \{\mathbf{A}_\sigma \in \Gamma\}, F)$, where

1. $S = \{s_1, \dots, s_n\}$ is the set of states, and s_1 is the start state,
2. \mathbf{A}_σ is the $n \times n$ real-valued stochastic transition matrix for symbol σ , that is, $\mathbf{A}_\sigma(i, j)$ is the value of the transition probability from state s_i to state s_j when reading symbol σ ,
3. $F \subseteq S$ is the set of accepting states.

The probability distribution of \mathcal{P} 's states at any point during the processing of the input string can be traced using an n -element row vector. For an input string $w \in \Sigma^*$, $\mathbf{w} = \mathfrak{c}w\mathfrak{s}$, $\mathbf{v}_0 = (1, 0, \dots, 0)_{1 \times n}$ denotes the initial state vector. The effect of reading the i th tape symbol can be calculated by multiplying the vector \mathbf{v}_{i-1} by the matrix \mathbf{A}_{w_i} , yielding \mathbf{v}_i . $\mathbf{v}_{|w|} = \mathbf{v}_0 \mathbf{A}_{w_1} \cdots \mathbf{A}_{w_{|w|}}$ denotes the final state vector. The acceptance probability of w by \mathcal{P} is

$$f_{\mathcal{P}}(w) = \sum_{s_i \in F} \mathbf{v}_{|w|}(i), \quad (1)$$

where $\mathbf{v}_{|w|}(i)$ denotes the i th entry of $\mathbf{v}_{|w|}$.

Definition 2. A generalized probabilistic finite automaton (GPFA) with $n \in \mathbb{Z}^+$ states is a 5-tuple $\mathcal{G} = (S, \Sigma, \{\mathbf{A}_\sigma \in \Sigma\}, \mathbf{v}_0, \mathbf{f})$, where

1. $S = \{s_1, \dots, s_n\}$ is the set of states,
2. \mathbf{A}_σ is the $n \times n$ real-valued transition matrix for symbol σ , that is, $\mathbf{A}_\sigma(i, j)$ is the (possibly negative) “weight” of the transition from state s_i to state s_j when reading symbol σ ,
3. \mathbf{v}_0 is the real-valued initial $1 \times n$ vector, and,
4. \mathbf{f} is the real-valued final $n \times 1$ vector.

A GPFA \mathcal{G} is associated by a function $f_{\mathcal{G}} : \Sigma^* \rightarrow \mathbb{R}$, in the following way: For an input string $w \in \Sigma^*$,

$$f_{\mathcal{G}}(w) = \mathbf{v}_0 \mathbf{A}_{w_1} \cdots \mathbf{A}_{w_{|w|}} \mathbf{f}. \quad (2)$$

Definition 3. A (1-way) Kondacs-Watrous quantum finite automaton (KWQFA) [20] with $n \in \mathbb{Z}^+$ states is a 5-tuple $\mathcal{M} = (Q, \Sigma, \{\mathbf{U}_\sigma \in \Gamma\}, Q_{acc}, Q_{rej})$, where

1. $Q = \{q_1, \dots, q_n\}$ is the set of states, and q_1 is the initial state,
2. \mathbf{U}_σ is the $n \times n$ complex-valued unitary transition matrix for symbol σ , that is, $\mathbf{U}_\sigma(j, i)$ is the amplitude of the transition from q_i to q_j when reading the symbol σ ,

3. Q_{acc} and Q_{rej} , disjoint subsets of Q , are the sets of accepting and rejecting states, and $Q_{non} = Q \setminus (Q_{acc} \cup Q_{rej})$ is the set of non-halting states.

The amplitude distribution of the states of a quantum automaton is represented by an n -element column vector. $|u_0\rangle$, the initial state vector, equals $(1, 0, \dots, 0)_{1 \times n}^T$. Note the difference with probabilistic automata.

For a given input string $w \in \Sigma^*$, \mathcal{M} scans the tape, containing $w = \$w\$$, from the left to the right. During the processing of each symbol, the machine undergoes two operations: First, its state vector evolves according to the unitary transformation associated with the scanned symbol, that is,

$$|u_i\rangle = U_{w_i}|u_{i-1}\rangle.$$

Then, the machine is observed to see whether it has accepted, rejected, or not halted yet. At this point, each accepting state with amplitude α adds $|\alpha|^2$ to the overall acceptance probability $f_{\mathcal{M}}(w)$ of the input². In the particular KWQFA's that will be described in this paper, intermediate observations can yield the result "reject", but the accepting states can be entered only at the end of the computation, after scanning the right end-marker $\$$. Halting states "drop out" of the state vector $|u\rangle_i$, their amplitudes being replaced with zeros, and the head moves on to the next symbol.

As we have defined them, PFA's process all of the input string before deciding on acceptance or rejection, whereas KWQFA's can halt before reaching the end of the input. (The QFA variant that precisely corresponds to Definition 1 is the *Moore-Crutchfield QFA* (MCQFA) [28].) This difference should not distract the reader, since it is easy to show that the classes of languages recognized by PFA's, both with general cutpoint, and with cutpoint 0, (to be defined in the next subsection,) do not change when the model is modified to give it this additional capability³. This is true for all PFA variants that may be obtained by appropriately reconfiguring Definition 1 to correspond to the various QFA models that are cited in this paper. The only crucial distinction between Definitions 1 and 3 is the one between classical and quantum.

2.2 Languages

Definition 4. An automaton \mathcal{A} defined over alphabet Σ divides Σ^* into three disjoint subsets with cutpoint $\lambda \in \mathbb{R}$:

1. $\mathbb{L}(\mathcal{A}, < \lambda) = \{w \in \Sigma^* \mid f_{\mathcal{A}}(w) < \lambda\}$,
2. $\mathbb{L}(\mathcal{A}, = \lambda) = \{w \in \Sigma^* \mid f_{\mathcal{A}}(w) = \lambda\}$,
3. $\mathbb{L}(\mathcal{A}, > \lambda) = \{w \in \Sigma^* \mid f_{\mathcal{A}}(w) > \lambda\}$.

Additionally, we define $\mathbb{L}(\mathcal{A}, \neq \lambda) = \mathbb{L}(\mathcal{A}, < \lambda) \cup \mathbb{L}(\mathcal{A}, > \lambda)$.

Definition 5. The pair (\mathcal{A}, λ) is equivalent under cutpoint separation to the pair (\mathcal{A}', λ') , denoted as $(\mathcal{A}, \lambda) \equiv (\mathcal{A}', \lambda')$, if

$$\begin{aligned}\mathbb{L}(\mathcal{A}, < \lambda) &= \mathbb{L}(\mathcal{A}', < \lambda') \\ \mathbb{L}(\mathcal{A}, = \lambda) &= \mathbb{L}(\mathcal{A}', = \lambda') \\ \mathbb{L}(\mathcal{A}, > \lambda) &= \mathbb{L}(\mathcal{A}', > \lambda'),\end{aligned}$$

² This is the behavior allowed by the general KWQFA definition.

³ Note that, when comparing two QFA variants with each other, this kind of difference is very important, since it usually affects the computational power of the models.

where $\mathcal{A}, \mathcal{A}'$ are automata and $\lambda, \lambda' \in \mathbb{R}$ are cutpoints.

Definition 6. The language recognized by automaton \mathcal{A} with cutpoint $\lambda \in \mathbb{R}$ is defined as

$$\mathbb{L}(\mathcal{A}, \lambda) = \mathbb{L}(\mathcal{A}, > \lambda).$$

$\mathbb{L}(\mathcal{A}, \lambda)$ is said to be recognized by automaton \mathcal{A} with one-sided cutpoint $\lambda \in \mathbb{R}$ if $\mathbb{L}(\mathcal{A}, < \lambda) = \emptyset$.

Definition 7. [32]

1. The languages recognized by PFA's with cutpoint $\lambda \in [0, 1)$ constitute the class of stochastic languages ($S^>$). The collection of languages whose complements are stochastic is the class $\text{co-}S^>$.
2. Languages of the form $\mathbb{L}(\mathcal{P}, = \lambda)$, for any PFA \mathcal{P} , and any $\lambda \in [0, 1]$, constitute the class $S^=$.
3. Languages of the form $\mathbb{L}(\mathcal{P}, \neq \lambda)$, for any PFA \mathcal{P} , and any $\lambda \in [0, 1]$, constitute the class exclusive stochastic languages (S^\neq).

Remark 1. In the study of complexity classes defined in terms of Turing machines, “recognition with cutpoint” is used synonymously with “unbounded-error recognition”. This usage does not provide an appropriate coverage of the intuitive concept of unbounded-error computation that we described in Section 1 in the case of PFA's: Given a PFA \mathcal{P} which recognizes a language L with cutpoint, one can build a new PFA \mathcal{P}' for the complement of L by just switching the statuses of the accepting and non-accepting states of \mathcal{P} . Since \mathcal{P}' accepts any member of \overline{L} with greater probability than any nonmember, we say that it recognizes \overline{L} with unbounded error. However, since $S^>$ is not known to be closed under complementation, we do not know in general whether \overline{L} is stochastic or not. For this reason, we take $S^> \cup \text{co-}S^>$ to be the class of languages recognized with unbounded error by PFA's⁴.

Definition 8.

1. The languages recognized by KWQFA's (MCQFA's) with cutpoint $\lambda \in [0, 1)$ constitute the class QL (MCL).
2. The languages recognized by KWQFA's (MCQFA's) with cutpoint 0, i.e., those of the form $\mathbb{L}(\mathcal{M}, 0)$, for any KWQFA (MCQFA) \mathcal{M} , constitute the class NQL (NMCL).

As mentioned before, nondeterministic computation corresponds to recognition with cutpoint 0, and so NMCL and NQL denote the classes of languages recognized by nondeterministic MCQFA's and KWQFA's, respectively.

⁴ Note that $S_{rat}^>$, the class of languages recognized with cutpoint $\frac{1}{2}$ by PFA's whose transition matrices contain only rational numbers, is known [37] to be closed under complementation. It is however customary to define PFA's and QFA's with general transition probabilities/amplitudes, as we did in Definitions 1 and 3, in the finite automata literature, and we follow this convention. See Section 6 for more on this point.

Fact 21. [36] Let \mathcal{G}_1 be a GPFA and $\lambda_1 \in \mathbb{R}$ be a cutpoint. For any cutpoint $\lambda_2 \in \mathbb{R}$, there exists a GPFA \mathcal{G}_2 such that $(\mathcal{G}_1, \lambda_1) \equiv (\mathcal{G}_2, \lambda_2)$.

Fact 22. [32] Let \mathcal{P}_1 be a PFA and $\lambda_1 \in [0, 1)$ be a cutpoint. For any cutpoint $\lambda_2 \in (0, 1)$, there exists a PFA \mathcal{P}_2 such that $(\mathcal{P}_1, \lambda_1) \equiv (\mathcal{P}_2, \lambda_2)$.

Fact 23. [43] For any PFA \mathcal{P} , there exists a KWQFA \mathcal{M} such that $(\mathcal{P}, \frac{1}{2}) \equiv (\mathcal{M}, \frac{1}{2})$.

Fact 24. [23,43] For any KWQFA \mathcal{M} and cutpoint $\lambda \in [0, 1)$, there exists a GPFA \mathcal{G} such that $(\mathcal{M}, \lambda) \equiv (\mathcal{G}, \lambda)$.

Fact 25. [36] For any GPFA \mathcal{G} and cutpoint $\lambda_1 \in \mathbb{R}$, there exist a PFA \mathcal{P} and a cutpoint $\lambda_2 \in (0, 1)$ such that $(\mathcal{G}, \lambda_1) \equiv (\mathcal{P}, \lambda_2)$.

Fact 26. [6] $\text{MCL} \subsetneq \text{S}^>$.

Fact 27. [8] Any MCQFA with n states can be simulated by a KWQFA with $2n$ states, so $\text{MCL} \subseteq \text{QL}$ and $\text{NMCL} \subseteq \text{NQL}$.

Fact 28. [32] The class of regular languages is a proper subset of both S^\neq and $\text{S}^=$.

Fact 29. [32] $\text{S}^\neq \subsetneq \text{S}^>$ and $\text{S}^> \setminus \text{S}^= \neq \emptyset$.

Fact 210. [9,29] The class of regular languages is a proper subset of NQL .

By Facts 23-25, $\text{QL} = \text{S}^>$, PFA's and KWQFA's have the same language recognition power with general cutpoint and with unbounded error (Remark 1). It has in fact been shown [42] that all one-way QFA models [31,11,7,18] that generalize the KWQFA are also equivalent to the PFA in this regard. (See Subsection 3.2 for more on this.)

We are interested in the case of one-sided unbounded error, where one of the two responses that the machine can output about the membership of the input string in the recognized language is correct with certainty, and the other response has a nonzero probability of being correct. We say that such an automaton has *positive one-sided error* if it rejects non-members of its language with certainty. This corresponds to recognition with cutpoint 0. The opposite case is called *negative one-sided error*, where the language in question is of the form $\mathbb{L}(\mathcal{A}, \neq 1)$, recalling that, when \mathcal{A} is a PFA or a QFA, $f_{\mathcal{A}}$ has range $[0, 1]$.

PFA's can recognize all and only the regular languages with cutpoint 0 [26]. KWQFA's can do more than that, as will be characterized in the next section.

3 Languages Recognized with One-sided Error

We start the presentation of our main result by stating a fact which will be useful in several proofs in the paper.

Lemma 1. *For any language L , $L \in S^\neq$ if and only if there exists a GPFA that recognizes L with one-sided cutpoint 0.*

Proof. The forward direction is proven on page 171 of [32]. In the reverse direction, if a GPFA recognizes L with one-sided cutpoint 0, then $L \in S^\neq$ by Fact 25. \square

3.1 A characterization of NQL

Lemma 2. $S^\neq \subseteq NQL$.

Proof. If $L \in S^\neq$, then there exists an n -state PFA $\mathcal{P} = (S, \Sigma, \{A_{\sigma \in \Gamma}\}, F)$ such that $L = \mathbb{L}(\mathcal{P}, \neq \frac{1}{2})$. We define S' , v'_0 , and $\{A'_{\sigma \in \Gamma}\}$ as follows:

1. $S' = S \cup \{s_{n+1}, s_{n+2}, s_{n+3}\}$;
2. $v'_0 = (1, 0, \dots, 0)$ is a $1 \times (n+3)$ -dimensional row vector;
3. Each A'_σ is a $(n+3) \times (n+3)$ -dimensional matrix:

$$A'_\Phi = \left(\begin{array}{c|ccc} \frac{1}{2}A_\Phi[r_1] & 0 & 0 & \frac{1}{2} \\ \hline 1 & 0 & \cdots & 0 \\ & \vdots & & \vdots \\ 1 & 0 & \cdots & 0 \end{array} \middle| \begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array} \right), \quad A'_{\sigma \in \Sigma} = \left(\begin{array}{c|ccc} A_\sigma & & & \\ \hline & 0_{n \times 3} & & \\ \hline & 1 & 0 & 0 \\ 0_{3 \times n} & 0 & 1 & 0 \\ & 0 & 0 & 1 \end{array} \right),$$

$$A'_\S = \left(\begin{array}{c|ccc} A_\S & & & \\ \hline & 0_{n \times 3} & & \\ \hline & 1 & 0 & 0 \\ 0_{3 \times n} & 0 & 1 & 0 \\ & 0 & 0 & 1 \end{array} \right) \left(\begin{array}{c|ccc} t_{1,1} & t_{1,2} & 0 & \\ \hline & \vdots & & \\ & t_{n,1} & t_{n,2} & 0 \\ \hline & 1 & 0 & 0 \\ 0_{3 \times n} & 0 & 1 & 0 \\ & \frac{-1}{2} & \frac{1}{2} & 0 \end{array} \right),$$

where $A_\Phi[r_1]$ is the first row of A_Φ ; $t_{i,1} = 1$ and $t_{i,2} = 0$ when $s_i \in F$, and $t_{i,1} = 0$ and $t_{i,2} = 1$ when $s_i \notin F$ for $1 \leq i \leq n$.

For a given input $w \in \Sigma^*$, $w = \phi w \$$, let $v'_{|w|} = v'_0 A'_\Phi A'_{w_1} \cdots A'_{w_{|w|}} A'_\S$. It is easily verified that this computation “imitates” the processing of w by \mathcal{P} ; the first n entries of the manipulated vector v' contain exactly the state vector of \mathcal{P} (multiplied by $\frac{1}{2}$) in the corresponding steps of its execution. The last matrix multiplication results in

$$v'_{|w|} = \left(0_{1 \times n} \middle| \frac{2f_{\mathcal{P}}(w) - 1}{4}, \frac{3 - 2f_{\mathcal{P}}(w)}{4}, 0 \right).$$

The $(n+1)^{st}$ entry of $\mathbf{v}'_{|w|}$ equals 0 if and only if $w \notin L$.

Using a modified version of the PFA simulation method described in [43], we can construct a KWQFA $\mathcal{M} = (Q, \Sigma, \{\mathbf{U}_{\sigma \in \Gamma}\}, Q_{acc}, Q_{rej})$ recognizing L with cutpoint 0: For each $\sigma \in \Gamma$, \mathbf{U}_{σ} is built according to the template

$$\mathbf{U}_{\sigma}^T = \left(\begin{array}{c|c|c} \mathbf{c}_{\sigma} \mathbf{A}'_{\sigma} & \mathbf{c}_{\sigma} \mathbf{B}_{\sigma} & \mathbf{c}_{\sigma} \mathbf{C}_{\sigma} \\ \hline & D_{\sigma} & \end{array} \right),$$

where $\mathbf{B}_{\sigma} = [b_{i,j}]$ is a lower triangular matrix, and $\mathbf{C}_{\sigma} = [c_{i,j}]$ is a diagonal matrix. The entries of \mathbf{U}_{σ} are computed iteratively using the following procedure:

1. The entries of \mathbf{B}_{σ} and \mathbf{C}_{σ} are set to 0.
2. The entries of \mathbf{B}_{σ} are updated to make the rows of $(\mathbf{A}'_{\sigma} \mid \mathbf{B}_{\sigma})$ pairwise orthogonal. Specifically,

for $i = 1 \cdots n+2$
 set $b_{i,i} = 1$
 for $j = i+1 \cdots n+3$
 set $b_{j,i}$ to some value so that the i th and j th rows become orthogonal
 set l_{max} to the maximum of the lengths (norms) of the rows of $(\mathbf{A}'_{\sigma} \mid \mathbf{B}_{\sigma})$

3. The diagonal entries of \mathbf{C}_{σ} are updated to make the length of each row of $(\mathbf{A}'_{\sigma} \mid \mathbf{B}_{\sigma} \mid \mathbf{C}_{\sigma})$ equal to l_{max} . Specifically,

for $i = 1 \cdots n+3$
 set l_i to the current length of the i th row of $(\mathbf{A}'_{\sigma} \mid \mathbf{B}_{\sigma})$
 set $c_{i,i}$ to $\sqrt{l_{max}^2 - l_i^2}$

4. Set c_{σ} to $\frac{1}{l_{max}}$.
5. The entries of \mathbf{D}_{σ} are selected to make \mathbf{U}_{σ}^T a unitary matrix. The transpose accounts for the difference between the probabilistic and quantum vector notations.

The state set $Q = Q_{non} \cup Q_{acc} \cup Q_{rej}$ is specified as:

1. $q_{n+1} \in Q_{acc}$ corresponds to state s_{n+1} ;
2. $q_{n+2} \in Q_{rej}$ corresponds to state s_{n+2} ;
3. $\{q_1, \dots, q_n, q_{n+3}\} \in Q_{non}$ correspond to the remaining states of S' , where q_1 is the start state;
4. All the new states that are defined during the construction of $\{\mathbf{U}_{\sigma \in \Gamma}\}$ are rejecting ones.

\mathcal{M} simulates the computation of \mathcal{P} for a given input string $w \in \Sigma^*$ by representing the probability of each state s_j by the amplitude of the corresponding state q_j ; specifically, this amplitude equals $c_{\mathbb{Q}} \left(\prod_{i=1}^k c_{w_i} \right) \times \mathbf{v}'(j)$ immediately after the $(k+1)^{st}$ step of the computation [43], where $k \leq |w|$. The transitions from the $2n+6$ states added during the construction of $\mathbf{U}_{\sigma \in \Gamma}$ for ensuring unitarity

do not interfere with this simulation, since the computation halts immediately on the “branches” where these states are entered. Therefore, the top $n + 3$ entries of the state vector of \mathcal{M} equal

$$c_{\mathbb{C}} \left(\prod_{i=1}^{|w|} c_{w_i} \right) c_{\mathbb{S}} \left(0_{1 \times n} \left| \frac{2f_{\mathcal{P}}(w) - 1}{4}, \frac{3 - 2f_{\mathcal{P}}(w)}{4}, 0 \right. \right)^{\top}$$

just before the last measurement on the right end-marker. Since the amplitude of the only accepting state is nonzero if and only if $w \in L$, L is recognized by \mathcal{M} with cutpoint 0. \square

Lemma 3. $NQL \subseteq S^{\neq}$.

Proof. By Fact 24, there exists a GPFA with one-sided cutpoint 0 for any member L of NQL . By Lemma 1, L is an exclusive stochastic language. \square

Theorem 1. $S^{\neq} = NQL$.

Corollary 1. $S^=$ is precisely the class of languages that can be recognized with negative one-sided error by KWQFA's.

The superiority of KWQFA's over PFA's in the one-sided error setting now follows from Fact 28. By Fact 29, there exist languages that KWQFA's can recognize with two-sided, but not one-sided error. The class of these languages is precisely $(S^> \cup \text{co-}S^>) \setminus (S^{\neq} \cup S^=)$ (Remark 2.1). Note that the above results also establish that the class of languages recognized by NQFA's is not closed under complementation (Fact 52).

3.2 More general QFA models

Several one-way QFA models (like [30,31,11,7,18], and the one-way version of the machines of [4],) that generalize the KWQFA have appeared in the literature. In the bounded-error case, some of these generalized machines recognize more languages than the KWQFA. We claim that the classes of languages recognized by the nondeterministic versions of all these automata are identical to each other, and they coincide with NQL .

We demonstrate this fact for one of the most general models, namely, the quantum finite automaton with ancilla qubits (QFA-A) [31], which can simulate all known one-way QFA models. Let us give the name QFA-A_0 to the class of languages recognized with cutpoint 0 by QFA-A's. For any QFA-A \mathcal{M} , there exists a GPFA that computes exactly the same acceptance probability function as \mathcal{M} [45], so $\text{QFA-A}_0 \subseteq S^{\neq}$ by Lemma 1. Since any KWQFA can be simulated by a QFA-A, $NQL \subseteq \text{QFA-A}_0$. Therefore, $\text{QFA-A}_0 = NQL = S^{\neq}$.

3.3 Space efficiency of QFA's with cutpoint 0

It is well known [3,27] that some infinite families of languages can be recognized with one-sided bounded error by just tuning the transition amplitudes of a QFA with a constant number of states, whereas the sizes of the corresponding PFA's grow without bound. After a simple example, we will argue that this advantage is also valid in the unbounded error case.

Definition 9. For $m \in \mathbb{Z}^+$, $L_m \subseteq \{a\}^*$ is defined as

$$L_m = \{a^i \mid i \bmod (m) \neq 0\}.$$

Theorem 2. For $m > 1$, L_m can be recognized by a 2-state MCQFA⁵ with cutpoint 0.

Proof. \mathcal{M} begins the computation at state q_0 , and each transition with the symbol a corresponds to a rotation⁶ by angle $\frac{\pi}{m}$ in the $|q_0\rangle$ - $|q_1\rangle$ plane, where q_1 is the accepting state. \square

For any positive n , it is known [26] that every n -state PFA with cutpoint 0 has an equivalent nondeterministic finite automaton with the same number of states. Therefore, only finitely many distinct languages can be recognized with one-sided unbounded error by PFA's with at most n states.

Combining this with the fact that any n -state PFA with cutpoint 0 can be simulated by a KWQFA with $2n + 4$ states using a simple adaptation of the technique of [43], the superiority of QFA's over PFA's in this regard is established⁷.

4 S^\neq , $S^=$, and Languages Recognized with Two-sided Error

To gain a better understanding of the classes of languages recognizable by positive one-sided, negative one-sided, and necessarily two-sided error by QFA's, we examine some examples from each of those families. Bertoni and Carpentieri [6] showed that L_{neq} is in NMCL, and that its complement, say, $L_{eq} = \{w \in \{a, b\}^* \mid |w|_a = |w|_b\}$, is not in MCL. Now that we have Theorem 1, we can use the well-known results [32,?] from the PFA literature that state that $L_{eq} \in S^=$, $L_{neq} \in S^\neq$, but not vice versa, to conclude that stronger QFA variants also can not recognize L_{eq} with positive one-sided error, and neither can they recognize L_{neq} with negative one-sided error. Similarly, Lāce *et al.* [21] proved recently that the complement of the palindrome language $L_{pal} = \{w \in \{a, b\}^* \mid w = w^r\}$ is in NQL. We can show the corresponding result for L_{pal} using the following fact:

Fact 41. [14] Let $L \in S^=$. Then there exists a natural number $n \geq 1$, such that for any strings $u, v, y \in \Sigma^*$,

$$\text{if } uv, uyv, \dots, uy^{n-1}v \in L, \text{ then } uy^*v \subseteq L.$$

Theorem 3. $L_{pal} \notin S^\neq$.

⁵ There is an equivalent 4-state KWQFA.

⁶ For details of a similar construction for a nonregular language, see [6].

⁷ Note that a QFA-A can realize this simulation with just n states.

Proof. Suppose that $L_{pal} \in S^\neq$. Then $\overline{L_{pal}} \in S^\equiv$. Let $u = a^n b$, $y = a$, and $v = \varepsilon$.

$$a^n b, a^n b a, \dots, a^n b a^{n-1} \in \overline{L_{pal}}$$

imply that $a^n b a^n \in \overline{L_{pal}}$ by Fact 41. Since this string is actually a member of L_{pal} , we have a contradiction. \square

We will now exhibit some languages which can only be recognized by two-sided error by a QFA.

Theorem 4. $L = \{aw_1 \cup bw_2 \mid w_1 \in L_{eq}, w_2 \in L_{neq}\} \in S^> \setminus (S^\equiv \cup S^\neq)$.

Proof. Suppose that $L \in S^\neq$, then there exists a GPFA

$$\mathcal{G} = (S, \Sigma, \{A_{\sigma \in \{a,b\}}\}, v_0, f)$$

recognizing L with one-sided cutpoint 0. The GPFA

$$\mathcal{G}' = (S, \Sigma, \{A_{\sigma \in \{a,b\}}\}, v_0 A_a, f)$$

recognizes L_{eq} with one-sided cutpoint 0, meaning that $L_{eq} \in S^\neq$. This contradicts the well-known fact mentioned in the first paragraph of this section. Suppose now that $L \in S^\equiv$, then

$$\overline{L} = \{\varepsilon \cup aw_2 \cup bw_1 \mid w_1 \in L_{eq}, w_2 \in L_{neq}\}$$

is in S^\neq , which also results in a contradiction for the same reason. Since both L_{eq} and its complement are stochastic, it is not difficult to show that L is stochastic. \square

Lemma 4. $L_{lt} = \{w \in \{a,b\}^* \mid |w|_a < |w|_b\} \notin (S^\equiv \cup S^\neq)$.

Proof. Suppose that $L_{lt} \in S^\equiv$. Let $u = \varepsilon$, $y = a$, and $v = b^n$.

$$b^n, ab^n, \dots, a^{n-1}b^n \in L_{lt}$$

imply that $a^n b^n \in L_{lt}$ by Fact 41. Since this string is actually a member of $\overline{L_{lt}}$, we have a contradiction.

Similarly, suppose that $L_{lt} \in S^\neq$, or $\overline{L_{lt}} \in S^\equiv$. Let $u = a^n$, $y = b$, and $v = b$.

$$a^n b, a^n b^2, \dots, a^n b^n \in \overline{L_{lt}}$$

imply that $a^n b^{n+1} \in \overline{L_{lt}}$ by Fact 41. Since this string is actually a member of L_{lt} , we have a contradiction. \square

Corollary 2. $L_{lt} \in S^> \setminus (S^\equiv \cup S^\neq)$.

Proof. This follows from Lemma 4 and the fact that $L_{lt} \in S^>$ [33,19]. \square

Theorem 5. $L_{eq \cdot b} = L_{eq} \cdot b^+ \in S^> \setminus (S^= \cup S^\neq)$.

Proof. The proof of $L_{eq \cdot b} \notin (S^= \cup S^\neq)$ uses the setup presented in Lemma 4, i.e.,

1. select $u = \varepsilon$, $y = a$, and $v = b^n$ to contradict with $L_{eq \cdot b} \in S^=$,
2. select $u = a^n$, $y = b$, and $v = b$ to contradict with $L_{eq \cdot b} \in S^\neq$.

Any string w is a member of $L_{eq \cdot b}$ if and only if it has the following three properties:

- w ends with b .
- $w \in L_{lt}$.
- Let u be the longest prefix of w ending with a ($u = \varepsilon$ if $w \in \{b^*\}$). Then, $u \in \overline{L_{lt}}$.

Since these properties can be checked easily by a two-way PFA with bounded error, $L_{eq \cdot b} \in S^>$ [33,19]. \square

We conclude this section by showing the stochasticity of an important family of languages.

Definition 10. [24] *The word problem for a group is the problem of deciding whether or not a product of group elements is equal to the identity element.*

Definition 11. Let $\mathcal{G}^k = (G, \circ)$ be a finitely generated free group with a basis

$$\Sigma = \{\sigma_1, \dots, \sigma_k, \sigma_1^{-1}, \dots, \sigma_k^{-1}\},$$

where $k \in \mathbb{Z}^+$ is the rank of \mathcal{G}^k . $L_{wp}(\mathcal{G}^k) \subseteq \Sigma^*$ is the language defined as

$$L_{wp}(\mathcal{G}^k) = \{w = w_1 \cdots w_{|w|} \mid w_i \in \Sigma, 1 \leq i \leq |w|, w_1 \circ \cdots \circ w_{|w|} = \iota\},$$

where $\iota \in G$ is the identity element of \mathcal{G}^k .

Fact 42. (Page 1 of [25]) Let $\mathcal{G}_1^{k_1}$ and $\mathcal{G}_2^{k_2}$ be finitely generated free groups. Then $\mathcal{G}_1^{k_1}$ and $\mathcal{G}_2^{k_2}$ are isomorphic if and only if $k_1 = k_2$.

Corollary 3. $L_{wp}(\mathcal{G}_1^{k_1})$ and $L_{wp}(\mathcal{G}_2^{k_2})$ are isomorphic if and only if $k_1 = k_2$, where $\mathcal{G}_1^{k_1}$ and $\mathcal{G}_2^{k_2}$ are finitely generated free groups.

As a generic name, $L_{wp}(k)$ can be used instead of $L_{wp}(\mathcal{G}^k)$ due to Corollary 3, where $k \in \mathbb{Z}^+$.

Fact 43. [39] $L_{wp}(1) \in S^>$.

Fact 44. [9] $L_{wp}(k) \in \text{co-NMCL}$, the class of languages whose complements are in NMCL, for any $k \in \mathbb{Z}^+$.

Corollary 4. $L_{wp}(k) \in S^=$ for any $k \in \mathbb{Z}^+$.

We will now provide a proof of the following theorem.

Theorem 6. $L_{wp}(k) \in S^>$ for any $k \geq 2$.

In fact, Theorem 6 was stated as a corollary on page 1463 of [9], but the purported proof there was based on the claim that $\text{co-NMCL} \subseteq \text{MCL} \subseteq S^>$. It is however known [6], as we mentioned above, that a member of co-NMCL (L_{eq}) lies outside MCL . Furthermore, the same demonstration can be easily extended to $L_{wp}(k)$, where $k \in \mathbb{Z}^+$.

Corollary 5. $L_{wp}(k) \notin \text{MCL}$ for any $k \in \mathbb{Z}^+$.

Since it is still an open problem whether $S^= \subseteq S^>$ or not, we cannot use Corollary 4 directly to prove Theorem 6. Instead, we will focus on a subclass of $S^=$ that is known to be a subset of $S^>$.

Definition 12. [37] $S_{rat}^=$ is the class of the languages of the form $\mathbb{L}(\mathcal{G}, = \lambda)$, where \mathcal{G} is a rational GPFA, (i.e. one whose transition matrices and initial and final vectors contain only rational numbers,) and λ is a rational number. Additionally, S_{rat}^\neq is the class of languages whose complements are in $S_{rat}^=$.

Fact 45. [37] $S_{rat}^= \subsetneq S^>$.

Definition 13. $\text{SO}_3(\mathbb{Q})$ is the group of rotations on \mathbb{R}^3 that are 3×3 dimensional orthogonal matrices having only rational entries with determinant +1.

Fact 46. [34,13] For any $k \geq 2$, $\text{SO}_3(\mathbb{Q})$ contains a free subgroup with rank k , namely \mathcal{S}^k .

Proof of Theorem 6. For $L_{wp}(k)$, we define a rational GPFA $\mathcal{G}_k = (\{s_1, s_2, s_3\}, \Sigma, \{A_\sigma \in \Sigma\}, v_0, f)$, where

1. $\Sigma = \{R_1, \dots, R_k, R_1^{-1}, \dots, R_k^{-1}\}$ is a basis of \mathcal{S}^k ;
2. $A_\sigma = \sigma$ for each $\sigma \in \Sigma$;
3. $v_0 = (1, 0, 0)$;
4. $f = (1, 0, 0)^T$.

It is obvious that $w \in L_{wp}(k)$ if and only if $A_1 \dots A_w = I_{3 \times 3}$ if and only if $f_{\mathcal{G}_k}(w) = v_0 A_1 \dots A_w f = 1$, where $w \in \Sigma^*$. Thus, $L_{wp}(k) \in S_{rat}^=$ by selecting the cutpoint as 1. We can conclude with Fact 45. \square

5 Closure Properties

The previously discovered closure properties of $S^>$, S^\neq and $S^=$ are listed below.

Fact 51.

1. $S^>$ is not closed under union and intersection [16,15,22].
2. $S^>$ is closed under union and intersection with a regular language [10,35].
3. $S^>$ is closed under reversal [36].
4. $S^>$ is not closed under concatenation, Kleene closure, and homomorphism [12,38].
5. $S^>$ is closed under complementation over unary alphabets [16].

Fact 52.

1. Both S^\neq and $S^=$ are closed under union and intersection [32].
2. Neither S^\neq nor $S^=$ is closed under complementation [14].
3. $S^>$ is closed under intersection with a member of S^\neq [32].

We will prove several new nontrivial closure properties of the “one-sided” classes S^\neq and $S^=$.

5.1 Dissimilar closure properties of S^\neq and $S^=$

The proofs of the next few theorems use the capability of GPFA’s to implement nondeterministic branching by just adding the transition matrices of the branches, and the nice properties of computation with one-sided cutpoint 0.

Theorem 7. *S^\neq is closed under concatenation.*

Proof. If $L_1, L_2 \in S^\neq$, then there exist two GPFA’s $\mathcal{G}_1 = (S', \Sigma, \{A'_{\sigma \in \Sigma}\}, v'_0, f')$ and $\mathcal{G}_2 = (S'', \Sigma, \{A''_{\sigma \in \Sigma}\}, v''_0, f'')$ such that L_1 and L_2 are recognized with one-sided cutpoint 0 by \mathcal{G}_1 and \mathcal{G}_2 , respectively. Let n_1 and n_2 be the sizes of the state sets S_1 and S_2 , respectively.

We construct a new GPFA $\mathcal{G} = (S, \Sigma, \{A_{\sigma \in \Sigma}\}, v_0, f)$ recognizing $L = L_1 L_2$ (concatenation of L_1 and L_2) with one-sided cutpoint 0. The details of \mathcal{G} are as follows:

1. The size of S is $n = n_1 + n_2$;
2. v_0 is a $1 \times n$ row vector,
 - (a) $v_0 = (v'_0 \mid v''_0)$ if ε (empty string) belongs to L_1 , and
 - (b) $v_0 = (v'_0 \mid 0_{1 \times n_2})$ if $\varepsilon \notin L_1$;
3. f is a $n \times 1$ column vector,
 - (a) $f = ((f')^T \mid (f'')^T)^T$ if $\varepsilon \in L_2$, and
 - (b) $f = (0_{1 \times n_1} \mid (f'')^T)^T$ if $\varepsilon \notin L_2$;
4. $\{A_{\sigma \in \Sigma}\}$ is the set of $n \times n$ matrices,

$$A_\sigma = \left(\begin{array}{c|c} A'_\sigma & X_\sigma \\ \hline 0_{n_2 \times n_1} & A''_\sigma \end{array} \right), \quad (3)$$

where X_σ is an $n_1 \times n_2$ matrix, defined as

$$X_\sigma = \left(\begin{array}{c|c|c} \underbrace{v_0''(1)A'_\sigma f'}_{\text{column 1}} & \underbrace{v_0''(2)A'_\sigma f'}_{\text{column 2}} & \cdots & \underbrace{v_0''(n_2)A'_\sigma f'}_{\text{column } n_2} \end{array} \right). \quad (4)$$

The idea behind the construction is that for a given input string $w \in \Sigma^*$, each prefix of w , say, $u \in \Sigma^*$ ($w = uv$), is checked for belonging to L_1 , and if so, the rest, v , is checked for belonging to L_2 .

\mathcal{G} simulates \mathcal{G}_1 in the first n_1 positions of its state vector. If \mathcal{G}_1 accepts an input prefix u ending with σ , the result of the multiplication between that $1 \times n_1$ row vector describing the distribution after processing the first $|u| - 1$ input symbols and the column vector $A'_\sigma f'$, that is, $f_{\mathcal{G}_1}(u)$, will be positive. Otherwise, $f_{\mathcal{G}_1}(u) = 0$. By Equations 3 and 4, the vector $f_{\mathcal{G}_1}(u) \cdot v_0''$ will be added to the last n_2 positions of \mathcal{G} 's state vector, meaning that \mathcal{G}_2 will run on the remainder v of the input. The contribution of this branch of the computation to $f_{\mathcal{G}}(w)$ is just the product of the value $f_{\mathcal{G}_2}(v)$ and the coefficient $f_{\mathcal{G}_1}(u)$, and will be positive if both substrings are accepted by the respective machines, and zero otherwise. Since \mathcal{G}_2 starts running in this manner in each step, its part of the overall state vector contains in general the sum of many $1 \times n_2$ vectors, multiplied by their respective coefficients, at any intermediate step of the computation. The cases where the empty string appears in L_1 or L_2 are handled appropriately.

In other words,

$$f_{\mathcal{G}}(w) = \sum_{uv=w} f_{\mathcal{G}_1}(u) f_{\mathcal{G}_2}(v),$$

and \mathcal{G} recognizes the concatenation of L_1 with L_2 . □

Theorem 8. S^\neq is not closed under concatenation.

Proof. L_{eq} and $\{b\}^+$ are in S^\neq , but $L_{eq} \cdot \{b\}^+ = L_{eq} \cdot \{b\}^+$ is not, due to Theorem 5. □

Theorem 9. S^\neq is closed under Kleene closure.

Proof. If $L \in S^\neq$, then there exists a GPFA $\mathcal{G} = (S, \Sigma, \{A_{\sigma \in \Sigma}\}, v_0, f)$ such that $L \cup \{\varepsilon\}$ is recognized by \mathcal{G} with one-sided cutpoint 0. Let n be the size of the state set S .

We construct a new GPFA $\mathcal{G}' = (S, \Sigma, \{A'_{\sigma \in \Sigma}\}, v_0, f)$ recognizing L^* (Kleene closure of L) with one-sided cutpoint 0. Each element of $\{A'_{\sigma \in \Sigma}\}$ is defined as

$$A'_\sigma = A_\sigma + X_\sigma, \quad (5)$$

where X_σ is an $n \times n$ matrix, defined as

$$X_\sigma = \left(\begin{array}{c|c|c} \underbrace{v_0(1)A_\sigma f}_{\text{column 1}} & \underbrace{v_0(2)A_\sigma f}_{\text{column 2}} & \cdots & \underbrace{v_0(n)A_\sigma f}_{\text{column } n} \end{array} \right). \quad (6)$$

For a given input string $w \in \Sigma^*$, $|w| = l$,

$$f_{\mathcal{G}'}(w) = v_0[A_{w_1} + X_{w_1}][A_{w_2} + X_{w_2}] \cdots [A_{w_l} + X_{w_l}]f,$$

and so

$$f_{\mathcal{G}'}(w) = \sum_{u_1 u_2 \cdots u_k = w} \left(\prod_{i=1}^k f_{\mathcal{G}}(u_i) \right),$$

where $1 \leq k \leq l$ and each $u_i \in \Sigma^*$. Therefore, if w can be divided, i.e., $w = u_1 \cdots u_k$, such that each $u_i \in L$ ($f_{\mathcal{G}}(u_i) > 0$), then $f_{\mathcal{G}'}(w) > 0$. On the other hand, if there is no such division, then $f_{\mathcal{G}'}(w) = 0$. \square

Lemma 5. *Let $L_{eq'} = \{w \in \{a, b\}^* \mid |w|_a + 1 = |w|_b\}$. Then, $L_{eq'}^+ = L_{lt}$.*

Proof. It is obvious that if $w \in L_{eq'}^+$, then $w \in L_{lt}$. If $w \in L_{lt}$, then $\exists k \in \mathbb{Z}^+$ such that $|w|_b = |w|_a + k$. Then, there must exist $k + 1$ indices, $i_0 = 0 < i_1 < i_2 < \cdots < i_k = |w|$, such that each prefix of w of length i_j has j more b 's than a 's, where $1 \leq j \leq k$. In other words, w can be partitioned into k consecutive substrings, u_1, u_2, \dots, u_k , satisfying

1. $w = u_1 u_2 \cdots u_k$, and,
2. $|u_j| = i_j - i_{j-1}$, that is, u_j begins with the $(i_{j-1} + 1)$ th symbol of w and ends with the i_j th symbol of w , where $1 \leq j \leq k$.

Since each u_j is a member of $L_{eq'}$, we can conclude that $w \in L_{eq'}^+$. \square

Theorem 10. *$S^=$ is not closed under Kleene closure.*

Proof. It can be shown easily that $L_{eq'}$ is in $S^=$. However, $L_{eq'}^*$, which is $L_{lt} \cup \{\varepsilon\}$ by Lemma 5, is not in $S^=$ due to Corollary 2. \square

Lemma 6. *Let $h : \Sigma \rightarrow \Sigma \setminus \{\kappa\}$ be a homomorphism such that*

$$h(\sigma) = \begin{cases} \sigma, & \sigma \neq \kappa \\ \varepsilon, & \sigma = \kappa \end{cases},$$

where κ is a specific symbol in Σ . If $L \subseteq \Sigma^$ is in S^\neq , then so is $h(L)$.*

Proof. Let $\mathcal{G} = (S, \Sigma, \{A_{\sigma \in \Sigma}\}, v_0, f)$ be the GPFA recognizing L with one-sided cutpoint 0, and let $\Sigma' = \Sigma \setminus \{\kappa\}$. For any $w \in h(L)$, there exists a $u \in L$, such that $h(u) = w$, i.e.,

$$u = \kappa^{c_0} w_1 \kappa^{c_1} \cdots \kappa^{c_{|w|-1}} w_{|w|} \kappa^{c_{|w|}}$$

for some nonnegative integer c_i 's, where $0 \leq i \leq |w|$. In fact, we can bound all c_i 's by a natural number, say n_L , due to Fact 41: Suppose that none of the strings in

$$\{\kappa^{c_0} w_1 \kappa^{c_1} \cdots \kappa^{c_{|w|-1}} w_{|w|} \kappa^{c_{|w|}} \mid 0 \leq i \leq |w|, 0 \leq c_i \leq n_L\}$$

are members of L (so they are all members of $\overline{L} \in S^=$).

Then, by using Fact 41,

$$\begin{aligned} \kappa^* w_1 \kappa^{c_1} \dots \kappa^{c_{|w|-1}} w_{|w|} \kappa^{c_{|w|}} &\subseteq \overline{L}, (1 \leq i \leq |w|, 0 \leq c_i \leq n_L) \\ \kappa^* w_1 \kappa^* \dots \kappa^{c_{|w|-1}} w_{|w|} \kappa^{c_{|w|}} &\subseteq \overline{L}, (2 \leq i \leq |w|, 0 \leq c_i \leq n_L) \\ &\vdots \\ \kappa^* w_1 \kappa^* \dots \kappa^* w_{|w|} \kappa^* &\subseteq \overline{L}. \end{aligned}$$

We conclude that $w \notin h(L)$, which is a contradiction.

Therefore, for any input string $w \in (\Sigma')^*$, we can simulate the computation of \mathcal{G} on some u 's, where each c_i is guessed nondeterministically from the set $\{0, 1, \dots, n_L - 1\}$. The following matrix can be defined to implement the nondeterministic branching of the computation:

$$X_\kappa = I + \sum_{j=1}^{n_L-1} A_\kappa^j.$$

By embedding X_κ in a convenient way in the definition of \mathcal{G} , we can get the GPFA

$$\mathcal{G}' = (S, \Sigma', \{A'_{\sigma \in \Sigma'} = A_\sigma X_\kappa\}, v_0 X_\kappa, f),$$

which recognizes $h(L)$ with one-sided cutpoint 0. Hence, $f_{\mathcal{G}'}(w)$ can be calculated as

$$f_{\mathcal{G}'}(w) = \sum_{u \in \{\kappa^{c_0} w_1 \kappa^{c_1} \dots \kappa^{c_{|w|-1}} w_{|w|} \kappa^{c_{|w|}}\}} f_{\mathcal{G}}(u)$$

for the input string $w \in (\Sigma')^*$, where $0 \leq c_i \leq n_L - 1$, and $0 \leq i \leq |w|$. Since the computation paths resulting in $u \notin L$ produce $f_{\mathcal{G}}(u) = 0$, $f_{\mathcal{G}'}(w) > 0$ is satisfied only when there is a computation path resulting in $u \in L$. \square

Lemma 7. *Let $h : \Sigma \rightarrow \Upsilon^*$ be a homomorphism such that $|h(\sigma)| > 0$ for all $\sigma \in \Sigma$. If $L \subseteq \Sigma^*$ is in S^\neq , then so is $h(L)$.*

Proof. Let $\mathcal{G} = (S, \Sigma, \{A_{\sigma \in \Sigma}\}, v_0, f)$ be the GPFA recognizing L with one-sided cutpoint 0. We will show that there exists a GPFA

$$\mathcal{G}' = (S', \Upsilon, \{A'_{\gamma \in \Upsilon}\}, v'_0, f')$$

recognizing $h(L)$ with one-sided cutpoint 0.

\mathcal{G}' runs \mathcal{G} on a nondeterministically chosen input $u = u_1 u_2 \dots u_{|u|} \in \Sigma^*$, while checking whether its own input string w matches $h(u_1)h(u_2) \dots h(u_{|u|})$ or not. For each such nondeterministic computation path, we have the following cases:

1. at least one of the matches fail, $h(u) \neq w$, then all entries representing the state vector of \mathcal{G} in this branch are set to zero,
2. all substitutions succeed, $h(u) = h(u_1)h(u_2) \dots h(u_{|u|}) = w$, then
 - (a) $f_{\mathcal{G}}(u) = 0$ for $u \notin L$, and

(b) $f_{\mathcal{G}}(u) > 0$ for $u \in L$.

$f_{\mathcal{G}'}(w)$ will again be defined as the summation over all computation paths, i.e., $f_{\mathcal{G}'}(w) = \sum_{\{u|h(u)=w\}} f_{\mathcal{G}}(u)$. Hence, $w \in h(L)$ only if there is at least one successful substitution, $f(u) = w$, and $u \in L$.

The technical details of \mathcal{G}' are as follows:

1. For each $\sigma \in \Sigma$, we will use a separate $1 \times |h(\sigma)||S|$ dimensional region in the state vector to trace the substitutions.

$$\left(\underbrace{(\dots \dots)}_{\sigma_1} \underbrace{(\dots \dots)}_{\sigma_2} \dots \underbrace{(\dots \dots)}_{\sigma_{|\Sigma|}} \right) \quad (7)$$

It is easily formulated that $|S'| = |S| (\sum_{\sigma \in \Sigma} |h(\sigma)|)$.

2. Each $\mathbf{A}'_{\gamma \in \Upsilon}$ is defined with respect to the separation above.

$$\mathbf{A}'_{\gamma} = \left(\begin{array}{c|c|c|c} \mathbf{A}'_{\gamma, \sigma_1} & T & \dots & T \\ \hline T & \mathbf{A}'_{\gamma, \sigma_2} & \dots & T \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline T & T & \dots & \mathbf{A}'_{\gamma, \sigma_{|\Sigma|}} \end{array} \right) \quad (8)$$

Each T is almost a zero matrix, except a case which will be described below.

3. Let $\sigma \in \Sigma$. Suppose that $h(\sigma) = \gamma_1 \gamma_2 \dots \gamma_l \in \Upsilon^*$ and $l > 0$. Then, the region corresponding to σ in \mathcal{G}' 's state vector can be partitioned to l $1 \times |S|$ blocks:

$$(\underbrace{\dots}_1 | \underbrace{\dots}_2 | \dots | \underbrace{\dots}_l). \quad (9)$$

For $\gamma \in \Upsilon$, $\mathbf{A}'_{\gamma, \sigma}$ can be partitioned into blocks of dimension $|S| \times |S|$: If $l = 1$,

$$\mathbf{A}'_{\gamma, \sigma} = (T_l), \quad (10)$$

and if $l > 1$,

$$\mathbf{A}'_{\gamma, \sigma} = \left(\begin{array}{c|c|c|c|c} 0 & T_1 & 0 & \dots & 0 \\ \hline 0 & 0 & T_2 & \dots & 0 \\ \hline \vdots & \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & 0 & \dots & T_{l-1} \\ \hline T_l & 0 & 0 & \dots & 0 \end{array} \right), \quad (11)$$

where for $\gamma_i = \gamma$, $T_i = I$, and $T_i = 0$ otherwise, ($0 < i < l$); for $\gamma_l = \gamma$, $T_l = \mathbf{A}_{\sigma}$, and $T_l = 0$ otherwise. Additionally, for $\gamma_l = \gamma$, the bottom-leftmost blocks of all T 's that are on the same row with $\mathbf{A}'_{\gamma, \sigma}$ in (8) are equal to \mathbf{A}_{σ} ; for $\gamma_l \neq \gamma$, all those blocks contain all 0's.

4.

$$\mathbf{v}'_0 = \left(\underbrace{(\mathbf{v}_0 | 0, \dots, 0)}_{\sigma_1} \underbrace{(\mathbf{v}_0 | 0, \dots, 0)}_{\sigma_2} \dots \underbrace{(\mathbf{v}_0 | 0, \dots, 0)}_{\sigma_{|\Sigma|}} \right), \quad (12)$$

$$\mathbf{f}' = \left(\underbrace{(\mathbf{f} | 0, \dots, 0)}_{\sigma_1} \underbrace{(\mathbf{f} | 0, \dots, 0)}_{\sigma_2} \dots \underbrace{(\mathbf{f} | 0, \dots, 0)}_{\sigma_{|\Sigma|}} \right)^{\mathbf{T}}. \quad (13)$$

When $\sigma \in \Sigma$ is being guessed, where $h(\sigma) = \gamma_1\gamma_2 \cdots \gamma_l \in \Upsilon^*$ and $l > 0$, the simulated state vector of \mathcal{G} is written in the first slot of (9). Whenever the matches succeed for $\gamma_1, \dots, \gamma_{l-1}$, the simulated state vector of \mathcal{G} is transferred to the next slot in (9); in any other case, it is set to 0. When γ_l is successfully substituted, this simulated state vector is updated as if \mathcal{G} has read the symbol σ , and the result is transferred anew in the first slots of all the regions corresponding to symbols in Σ ; otherwise, it is set to 0. \square

Theorem 11. S^\neq is closed under homomorphism.

Proof. Let $h : \Sigma \rightarrow \Upsilon^*$ be a homomorphism, $L \subseteq \Sigma^*$, and $L \in S^\neq$. If h is a homomorphism of the form in Lemma 7, then the proof is complete.

Otherwise, suppose that there are $k \geq 1$ symbols in Σ , i.e., $\sigma_1, \sigma_2, \dots, \sigma_k$, such that $h(\sigma_i) = \varepsilon$, where $1 \leq i \leq k$. So, we can define k homomorphisms of the form in Lemma 6:

$$\begin{aligned} h_1 : \Sigma &\rightarrow \Sigma_1, \text{ where } \Sigma_1 = \Sigma \setminus \{\sigma_1\} \\ h_2 : \Sigma_1 &\rightarrow \Sigma_2, \text{ where } \Sigma_2 = \Sigma_1 \setminus \{\sigma_2\} \\ &\vdots \\ h_k : \Sigma_{k-1} &\rightarrow \Sigma_k, \text{ where } \Sigma_k = \Sigma_{k-1} \setminus \{\sigma_k\}. \end{aligned}$$

Additionally, we define $h_{k+1} : \Sigma_k \rightarrow \Upsilon^*$, where $h_{k+1}(\sigma) = h(\sigma)$ for $\sigma \in \Sigma_k$. Since h is the composition of the h_i 's ($1 \leq i \leq k+1$), $h(L)$ is also in S^\neq (Lemma 6 and 7). \square

Theorem 12. $S^=$ is not closed under homomorphism.

Proof. Consider the languages $L_1 = \{w_1cw_2 \mid w_1 \in L_{eq}, w_2 \in b^+\}$ and $L_2 = \{w_1cw_2 \mid w_1 \in L_{eq}, w_2 \in b^*\}$. It is not hard to show that both languages are in $S^=$. Let h_1 and h_2 be two homomorphisms defined as

- $h_1(a) = a, h_1(b) = b, h_1(c) = \varepsilon$, and
- $h_2(a) = a, h_2(b) = b, h_2(c) = b$.

$L_{eq \cdot b} = h_1(L_1) = h_2(L_2)$, and so $S^=$ is not closed under (ε -free) homomorphism due to Theorem 5. \square

5.2 Common closure properties of S^\neq and $S^=$

Theorem 13. S^\neq and $S^=$ are closed under inverse homomorphism.

Proof. Let $h : \Sigma \rightarrow \Upsilon^*$ be a homomorphism, $L \subseteq \Upsilon^*$, and $L \in S^\neq$, such that the GPFA $\mathcal{G} = (S, \Upsilon, \{A_{\gamma \in \Upsilon}\}, v_0, f)$ recognizes L with one-sided cutpoint 0. It is easily verified that

$$\mathcal{G}' = (S, \Sigma, \{A'_{\sigma \in \Sigma}\}, v_0, f),$$

where

$$A'_{\sigma \in \Sigma} = \begin{cases} A_{u_1} \cdots A_{u_{|h(\sigma)|}} & , h(\sigma) = u_1 \cdots u_{|h(\sigma)|} \neq \varepsilon \\ I & , h(\sigma) = \varepsilon \end{cases},$$

recognizes $h^{-1}(L)$ with one-sided cutpoint 0. The same setup can be extended to any language in $S^=$. \square

Theorem 14. S^\neq and $S^=$ are closed under reversal.

Proof. We use the same idea as [36]. If $L \in S^\neq$, then there exists a GPFA $\mathcal{G} = (S, \Sigma, \{A_{\sigma \in \Sigma}\}, v_0, f)$ such that L is recognized by \mathcal{G} with one-sided cutpoint 0. It is easily seen that $\mathcal{G}' = (S, \Sigma, \{A_{\sigma \in \Sigma}^T\}, f^T, v_0^T)$ recognizes the reverse of L with one-sided cutpoint 0. The same setup can be extended to any language in $S^=$. \square

Theorem 15. S^\neq and $S^=$ are closed under word quotient.

Proof. If $L \in S^\neq$, then there exists a GPFA $\mathcal{G} = (S, \Sigma, \{A_{\sigma \in \Sigma}\}, v_0, f)$ such that L is recognized by \mathcal{G} with one-sided cutpoint 0. For any given $w \in \Sigma^*$,

1. GPFA $\mathcal{G}_1 = (S, \Sigma, \{A_{\sigma \in \Sigma}\}, v_0 A_{w_1} \cdots A_{w_{|w|}}, f)$ recognizes the language $\{y \mid wy \in L\}$ with one-sided cutpoint 0;
2. GPFA $\mathcal{G}_2 = (S, \Sigma, \{A_{\sigma \in \Sigma}\}, v_0, A_{w_1} \cdots A_{w_{|w|}} f)$ recognizes the language $\{z \mid zw \in L\}$ with one-sided cutpoint 0.

The same setup can be extended to any language in $S^=$. \square

Theorem 16. S^\neq and $S^=$ are not closed under difference.

Proof. There exists an $L \in S^\neq$ such that $\overline{L} \notin S^\neq$.

1. Σ^* and L are in S^\neq , but $\Sigma^* \setminus L = \overline{L}$ is not;
2. Σ^* and \overline{L} are in $S^=$, but $\Sigma^* \setminus \overline{L} = L$ is not.

\square

Theorem 17. S^\neq and $S^=$ are closed under difference with a regular language.

Proof. Regular languages are closed under complementation, and S^\neq and $S^=$ are closed under intersection. \square

For completeness, we list below the following easy facts about MCL and NMCL:

1. NMCL is closed under both union and intersection.
2. Neither MCL nor NMCL is closed under complementation [6].
3. Both MCL and NMCL are closed under inverse homomorphism [28].
4. Both MCL and NMCL are closed under word quotient [9].

6 Concluding Remarks

In this paper, we gave a full characterization of the class of languages recognized by all QFA models which are at least as powerful as the Kondacs-Watrous QFA with cutpoint 0. This is the only known case where the language recognition power of one-way QFA's has been proven to be strictly greater than that of their probabilistic counterparts⁸. The superiority of QFA's over PFA's with regard to space efficiency in this setting was demonstrated. We also examined the limitations of recognition with one-sided error for these models. Several new closure properties of the related classes S^\neq and $S^=$ were proven.

The relationship between nondeterministic quantum complexity classes and counting classes has been studied in detail. It is known that $\text{NQP} = \text{co-C=P}$ [46]. More relevantly for our work, Watrous [40] has shown that $\text{NQSPACE}(s) = \text{co-C=SPACE}(s)$ for $s = \Omega(\log(n))$. Note that the subset of S^\neq defined using PFA's that only contain efficiently computable transition probabilities and cutpoint $\frac{1}{2}$ equals $\text{co-C=SPACE}(1)$, so we have proven⁹ that $\text{co-C=SPACE}(1) \subseteq \text{NQSPACE}(1)$, and whether the inclusion is strict or not depends on whether a two-way head would increase the computational power of a NQFA¹⁰.

The superiority of NQFA's over classical NFA's has ramifications about relationships among classical and quantum nondeterministic space complexity classes for all sublogarithmic bounds. Although the following result follows from a combination of previously known facts, we have not seen it stated anywhere:

Theorem 18. $\text{NSPACE}(s) \subsetneq \text{NQSPACE}(s)$ for $s = o(\log(n))$.

Proof. (Sketch.) Quantum Turing machines can simulate probabilistic Turing machines easily for any common space bound [41]. There exists a NQFA with efficiently computable amplitudes (i.e. a constant-space nondeterministic quantum Turing machine) which recognizes the language $L_{\text{neq}} = \{w \in \{a, b\}^* \mid |w|_a \neq |w|_b\}$ [6]. It is easily seen that L_{neq} is a nonregular deterministic context-free language (DCFL). It is known that no nonregular DCFL is in $\text{NSPACE}(s)$ for $s = o(\log(n))$ [1]. \square

For space bounds $s \in \Omega(\log(n))$, all we know in this regard is the trivial fact that $\text{NSPACE}(s) \subseteq \text{NQSPACE}(s)$ [40].

The succinct QFA models alluded to in Section 3.3 form the basis of a demonstration [44] of the fact that two-way QFA's can have a similar state complexity advantage over both their one-way versions, and two-way classical nondeterministic automata.

One important QFA variant that was not considered in this paper is the Latvian QFA [2], which is a generalization of the MCQFA not thought to be as powerful as the KWQFA. An examination of the corresponding classes for this model would be interesting.

Some other open questions related to this work are listed below.

⁸ From a “pedagogical” point of view, this seems to us to be one of the simplest setups in which a quantum computational model can be demonstrated to outperform the corresponding probabilistic model.

⁹ All our proofs stand when the transition probabilities and amplitudes are restricted to be efficiently computable numbers, as mentioned in [5]. We can in fact prove that the collection of languages recognized by the most general model of NQFA's [31] is precisely the class S_{rat}^\neq (see Definition 12) when all the amplitudes of the NQFA are restricted to be rational numbers.

¹⁰ For any two-way PFA \mathcal{M} and cutpoint $\lambda_1 \in [0, 1)$, there exist a one-way PFA \mathcal{P} and a cutpoint $\lambda_2 \in [0, 1)$ such that $(\mathcal{M}, \lambda_1) \equiv (\mathcal{P}, \lambda_2)$ [19], whereas two-way QFA's are more powerful than one-way QFA's in the general unbounded error setting [43].

Open Problem 1. Is MCL closed under union? Intersection?

Open Problem 2. Do NMCL and MCL coincide?

Open Problem 3. Does $S^{\neq} \cap S^=$ contain a nonregular language?

Open Problem 4. Is S^{\neq} countable or uncountable?

Open Problem 5. Is $S^>$ closed under complementation? (page 158 of [32])

Open Problem 6. Is $S^=$ a subset of $S^>$? (page 173 of [32])

Open Problem 7. Can NQFA's with a two-way tape head recognize more languages than the one-way model discussed here?

Acknowledgments

We are grateful to Andris Ambainis, John Watrous, and Flavio D'Alessandro for their helpful comments on the subject matter of this paper. We also thank Rūsiņš Freivalds for kindly providing us a copy of reference [21].

References

1. Helmut Alt, Viliam Geffert, and Kurt Mehlhorn. A lower bound for the nondeterministic space complexity of context-free recognition. *Information Processing Letters*, 42(1):25–27, 1992.
2. Andris Ambainis, Martin Beaudry, Marats Golovkins, Arnolds Ķikusts, Mark Mercer, and Denis Thérien. Algebraic results on quantum automata. *Theory of Computing Systems*, 39(1):165–188, 2006.
3. Andris Ambainis and Rūsiņš Freivalds. 1-way quantum finite automata: strengths, weaknesses and generalizations. In *FOCS'98: Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, pages 332–341, Palo Alto, California, 1998.
4. Andris Ambainis and John Watrous. Two-way finite automata with quantum and classical states. *Theoretical Computer Science*, 287(1):299–311, 2002.
5. Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5), 1997.
6. Alberto Bertoni and Marco Carpentieri. Analogies and differences between quantum and stochastic automata. *Theoretical Computer Science*, 262(1-2):69–81, 2001.
7. Alberto Bertoni, Carlo Mereghetti, and Beatrice Palano. Quantum computing: 1-way quantum automata. In Zoltán Ésik and Zoltán Fülöp, editors, *Developments in Language Theory*, volume 2710 of *LNCS*, pages 1–20. Springer, 2003.
8. Alex Brodsky. Models and characterizations of 1-way quantum finite automata. Master's thesis, The University of British Columbia, 1998.
9. Alex Brodsky and Nicholas Pippenger. Characterizations of 1-way quantum finite automata. *SIAM Journal on Computing*, 31(5):1456–1478, 2002.
10. R. G. Bukharaev. Theory of probabilistic automata. *Kibernetika*, (2):6–23, 1968.

11. Massimo Pica Ciamarra. Quantum reversibility and a new model of quantum automaton. In *FCT '01: Proceedings of the 13th International Symposium on Fundamentals of Computation Theory*, pages 376–379, London, UK, 2001. Springer-Verlag.
12. Stephen N. Cole. Real-time computation by n-dimensional iterative arrays of finite-state machines. *IEEE Transactions on Computers*, 18(4):349–365, 1969.
13. Flavio D'Alessandro and Alessandro D'Andrea. A non-commutativity statement for algebraic quaternions. *International Journal of Algebra and Computation*, 16(3):583–602, 2006.
14. Phan Dinh Diêu. On a class of stochastic languages. *Mathematical Logic Quarterly*, 17(1):421–425, 1971.
15. Michel Fliess. Automates stochastiques et séries rationnelles non commutatives. In *Automata, Languages, and Programming*, pages 397–411, 1973.
16. Michel Fliess. Propriétés booléennes des langages stochastiques. *Mathematical Systems Theory*, 7(4):353–359, 1973.
17. Mika Hirvensalo. Improved undecidability results on the emptiness problem of probabilistic and quantum cut-point languages. In *SOFSEM 2007: Theory and Practice of Computer Science*, volume 4362 of *Lecture Notes in Computer Science*, pages 309–319. Springer Berlin / Heidelberg, 2007.
18. Mika Hirvensalo. Various aspects of finite quantum automata. In *DLT '08: Proceedings of the 12th international conference on Developments in Language Theory*, pages 21–33, Berlin, Heidelberg, 2008. Springer-Verlag.
19. Jānis Kaneps. Stochasticity of the languages acceptable by two-way finite probabilistic automata. *Diskretnaya Matematika*, 1:63–67, 1989. (Russian).
20. Attila Kondacs and John Watrous. On the power of quantum finite state automata. In *FOCS'97: Proceedings of the 38th Annual Symposium on Foundations of Computer Science*, pages 66–75, Miami, Florida, 1997.
21. Lelde Lāce, Oksana Scegulnaja-Dubrovskaja, and Rūsiņš Freivalds. Languages recognizable by quantum finite automata with cut-point 0. presented at the 35th International Conference on Current Trends in Theory and Practice of Computer Science, SOFSEM, 2009.
22. Jānis Lapiņš. On nonstochastic languages obtained as the union and intersection of stochastic languages. *Avtom. Vychisl. Tekh.*, (4):6–13, 1974. (Russian).
23. Lvzhou Li and Daowen Qiu. Determining the equivalence for one-way quantum finite automata. *Theoretical Computer Science*, 403(1):42–51, 2008.
24. Richard J. Lipton and Yechezkel Zalcstein. Word problems solvable in logspace. *Journal of the ACM*, 24(3):522–526, 1977.
25. Roger C. Lyndon and Paul E. Schupp. *Combinatorial Group Theory*. Springer-Verlag, 1977.
26. Ioan Macarie. Closure properties of stochastic languages. Technical report, University of Rochester, Rochester, NY, USA, 1993.
27. Carlo Mereghetti, Beatrice Palano, and Giovanni Pighizzini. Note on the succinctness of deterministic, nondeterministic, probabilistic and quantum finite automata. *Theoretical Informatics and Applications*, 35(5):477–490, 2001.
28. Cristopher Moore and James P. Crutchfield. Quantum automata and quantum grammars. *Theoretical Computer Science*, 237(1-2):275–306, 2000.
29. Masaki Nakanishi, Takao Indoh, Kiyoharu Hamaguchi, and Toshinobu Kashiwabara. On the power of non-deterministic quantum finite automata. *IEICE Transactions on Information and Systems*, E85-D(2):327–332, 2002.
30. Ashwin Nayak. Optimal lower bounds for quantum automata and random access codes. In *FOCS'99: Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, pages 369–376, Washington, DC, USA, 1999. IEEE Computer Society.
31. Kathrin Paschen. Quantum finite automata using ancilla qubits. Technical report, University of Karlsruhe, 2000.
32. Azaria Paz. *Introduction to Probabilistic Automata*. Academic Press, New York, 1971.
33. Bala Ravikumar. Some observations on 2-way probabilistic finite automata. In *Proceedings of the 12th Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 392–403, London, UK, 1992. Springer-Verlag.
34. S. Świerczkowski. A class of free rotation groups. *Indagationes Mathematicae*, 5(2):221–226, 1994.
35. Paavo Turakainen. On stochastic languages. *Information and Control*, 12(4):304–313, 1968.
36. Paavo Turakainen. Generalized automata and stochastic languages. *Proceedings of the American Mathematical Society*, 21:303–309, 1969.
37. Paavo Turakainen. On languages representable in rational probabilistic automata. *Annales Academiae Scientiarum Fennicae, Ser.A*, (439):4–10, 1969.
38. Paavo Turakainen. Some closure properties of the family of stochastic languages. *Information and Control*, 18(3):253–256, 1971.

39. Paavo Turakainen. On nonstochastic languages and homomorphic images of stochastic languages. *Information Sciences*, 24(3):229–253, 1981.
40. John Watrous. Space-bounded quantum complexity. *Journal of Computer and System Sciences*, 59(2):281–326, 1999.
41. John Watrous. Quantum computational complexity. In Robert A. Meyers, editor, *Encyclopedia of Complexity and Systems Science*, pages 7174–7201. Springer, 2009.
42. Abuzer Yakaryilmaz and A. C. Cem Say. Language recognition by generalized quantum finite automata with unbounded error. In *4th Workshop on Theory of Quantum Computation, Communication, and Cryptography, TQC2009*, Waterloo, Ontario, Canada, 2009.
43. Abuzer Yakaryilmaz and A. C. Cem Say. Languages recognized with unbounded error by quantum finite automata. In *CSR'09: Proceedings of the Fourth International Computer Science Symposium in Russia*, volume 5675 of *Lecture Notes in Computer Science*, pages 356–367, 2009.
44. Abuzer Yakaryilmaz and A. C. Cem Say. Succinctness of two-way probabilistic and quantum finite automata. Technical Report arXiv:0903.0050v2, 2009. A preliminary version of this paper was presented at the AutoMathA Plenary Conference 2009, in Liège, Belgium.
45. Abuzer Yakaryilmaz and A. C. Cem Say. Unbounded-error quantum computation with small space bounds. in preparation, 2010.
46. Tomoyuki Yamakami and Andrew Chi-Chih Yao. $\text{NQP}_{\mathbb{C}} = \text{co-C=P}$. *Information Processing Letters*, 71(2):63–69, 1999.